# Utility Codes:
# Version 2000-1 Summary

TART is distributed with a number of utility codes that are designed to edit TART output for criticality or source problems. The primary reference to these utility codes remains Chapter 8: Utility Codes, of the TART95 documentation (included in the TART on-line documentation). Below is a brief introduction and update for each code.

## Changes in TART2000 Output Format

In order to accommodate many more zones the TART output format has been changed starting with TART2000; earlier only three digit zone number were accommodated by the format; this has been extended to five digits.

As a consequence all of the TART utility codes distributed with TART2000 CD have been updated to recognize the new output format. **WARNING** - do not use older versions of the TART utility codes; they have been completely superseded.

## Multiprocessing

**WARNING** - Currently MULTIPRO and TARTSUM can only be used for source, not criticality, problems.

**MULTIPRO**
Prepare input to start any number of TART runs. After ALL runs have finished, TARTSUM can be used to average the results of any number of runs.
**TARTSUM**
Combine results of a number of TART runs. This can be used to improve the statistical accuracy of your results by adding additional results. When using multiprocessing this can quickly improve results.

## Criticality Problems

**CRITEDIT**
Summarize static reactivity results in tabular form.
**BALANCE**
Edit criticality production, absorption and leakage spectra for an entire system to PLOTTAB input format. Use PLOTTAB to see your results.
**PATHC**
Edit pathlength (flux) spectra for individual spatial zones; PATHC = Path for Criticality Problems. Use PLOTTAB to see your results.

## Source Problems

**FLUXEDIT**
Edit flux and energy deposition for each spatial zone to PLOTTAB input format. Use TARTCHEK to overlay the results on your geometry.
**PATHS**
Edit pathlength (flux) spectra for individual spatial zones. Use PLOTTAB to see your results; PATHS = Path for Source Problems; neutrons and/or photon problems.
**EDIT1112**
Edit binary output to tabular form.
**PLOT1112**
Edit binary output to PLOTTAB input format. Use PLOTTAB to see your results.

## Input Parameters

If there are input parameters, they will be in a file with the same name as the utility codes, plus an extension of .INP, e.g., for the BALANCE code, BALANCE.INP. Each input file contains a complete, up-to-date description of all input parameters. As such, input parameters will not be described here.

### Input TART Results

TART.OUT is read by CRITEDIT, BALANCE, PATHC, PATHS, and FLUXEDIT. This is the TART default file name for TART output.

TART.OUT is written by TARTSUM. It reads a number of TART output files, and creates TART.OUT, containing the combined results.

EDIT1112 and PLOT1112 read binary output files created using TART tally type 11 or 12. These are TART options to score particles entering a zone and output their coordinates to a binary file for later use. Starting with the 2000 distribution, tally type 11 or 12 can alternately be used to tally absorptions within zones, rather than particles entering a zone, allowing for improved simulation of detectors.

## Multiprocessing

**WARNING** - Currently MULTIPRO and TARTSUM can only be used for source, not criticality, problems.

### MULTIPRO

MULTIPRO prepares input to be used to start any number of TART source problems, and to later combine the results using TARTSUM. This can be used to improve the statistical accuracy of your results by adding additional results. When using multiprocessing this can quickly improve results.

MULTIPRO reads ONE TART input problem, named TART.IN, and produces ANY number of copies of the input deck where each copy differs from the original ONLY by the random number sequence used (sentl 12) and binary tally type name (sentl 51).

MULTIPRO also produces a batch file named TART2000.BAT that can be used to start all of the input decks that it created. The contents of TART2000.BAT look like this,

```
tart2000-1 IN000001 OUT00001 > LST00001 &
tart2000-1 IN000002 OUT00002 > LST00002 &
tart2000-1 IN000002 OUT00002 > LST00003 &
                    .
                    .
                    .
tart2000-1 IN000049 OUT00049 > LST00049 &
tart2000-1 IN000050 OUT00050 > LST00050 &
```

Where in this example,
1) The TART code, tart2000-1 is used to simultaneously execute 50 problems.
2) IN000001 through IN000050 are 50 copies of the TART input (created by MULTIPRO)
3) OUT0001 through OUT00050 are 50 TART output listings (created by TART)
4) LST00001 through LST00050 are 50 dummy files for normal screen output (created by TART)

On UNIX and LINUX systems the & at the end of each line will run each problem in the background, thereby releasing your terminal to immediately continue on to start the next run, without delaying until the preceding run is completed. Essentially all 50 runs will be started at the same time.

On other systems, such as IBM-PC/Windows, currently & is ignored, and the problems are run sequentially one after another. Therefore on other systems the advantage of the simultaneous multiprocessing possible on UNIX and LINUX systems is currently not possible. However, note that the procedures described here can still be used to advantage

by using any number of IBM-PC/Windows computers to simultaneously run problems for you, and to then use TARTSUM to add the results together.

MULTIPRO also produces an input file that can be used with TARTSUM to add all of the results together. For example, in the case described above where 50 problems are run, MULTIPRO will produce a file named TARTSUM.INP. This file will contain the filenames of the 50 TART output listings to add together. For the above example this file would contain,

OUT000001
OUT000002
OUT000003
.
.
.
OUT000049
OUT000050

Input Parameters: **MULTIPRO.INP**

To use MULTIPRO, edit **MULTIPRO.INP** to define how many problems you want to run, and which random number sequence to start with; each successive problem will be assigned the next random number sequence is sequential order, and type,

MULTIPRO

On any UNIX or LINUX system when MULTIPRO finishes follow these steps,
1) Make sure TART2000.BAT is executable [chmod 777 TART2000.BAT] and execute it to start all of the problems running.
2) Use "top" to monitor execution of the problems.
3) When ALL of the problems have finished execute TARTSUM to average all of your results together and produce one final TART output file named TART.OUT.

It is as simple as that. Used properly this can be an extremely simple and yet powerful tool that can greatly accelerate how much work you can accomplish in a short period of time. For example, if you have available one or more of the currently available computers that have thousands of processors, you could use all of the processors to accelerate your work by factors of thousands.

## TARTSUM

TARTSUM will combine results of a number of TART runs. This can be used to improve the statistical accuracy of your results by adding additional results. When using multiprocessing this can quickly improve results.

The TARTSUM code reads a series of entire TART output files, for **any number of source problems**, and creates a combined file named TART.OUT. All of the output files

MUST correspond to EXACTLY the same TART input problem(s) - the ONLY thing that you are allowed to change is the random number sequence (sentl 12) [to make the results statistically independent], and binary tally type name (sentl 51)..

Input Parameters: **TARTSUM.INP**

To use TARTSUM, edit **TARTSUM.INP** to define the output you want, be sure that all of the files you want it to read are in the same directory, and that NONE are named TART.OUT (the combined output file name), and type,

TARTSUM

You will obtain results in a file named TART.OUT, that is in EXACTLY the same format as any other TART output file. Therefore, if you have any codes that process TART output - not to worry - they will work on the combined file, just as they work on any other TART output files.

# Criticality Problems

## CRITEDIT

CRITEDIT reads an entire TART output file, TART.OUT, for **any number of criticality problems**, and produces a table summarizing the results of the calculations.

Input parameters: **NONE**

To use CRITEDIT, be sure that the file you want it to read is named TART.OUT, and type,

CRITEDIT

You will obtain results both on your screen as the code runs, and in an output file named CRITEDIT.LST (CRITEDIT list).

There are two primary use for CRITEDIT, both are illustrated in the TART95 documentation,
1) Summarize the results of a variety of different criticality calculations.
2) Summarize the results of running the same criticality calculation a number of times. This can be used to further check on and improve the statistical accuracy of results

## BALANCE

For any criticality calculation the multiplication of the system (K-eff), is defined as a BALANCE between the **production** of neutrons and loss from the system due to **absorption and leakage**. BALANCE will allow you to see the results for any system as a function of neutron energy. Results include a spectrum of the incident energy at which neutron interactions produced neutrons (not the fission spectrum produced), as well as spectra of the energy at which neutrons were absorbed or leaked from the system. By comparing these three curves it is very simple to see what energy ranges are important. For example, those energy ranges where production exceeds the sum of absorption and leakage will tend to make a system super-critical, and those energy ranges where production is less than the sum of absorption and leakage will tend to make it sub-critical. The integral over energy of these results defines the multiplication of the system (K-eff).

The BALANCE code reads an entire TART output file, TART.OUT, for **one criticality problem**, and produces output in the PLOTTAB input format, so that you can immediately see your results.

Input Parameters: **BALANCE.INP**

To use BALANCE, edit **BALANCE.INP** to define the output you want, be sure that the file you want it to read is named TART.OUT, and type,

BALANCE

You will obtain results in a file named PLOTTAB.CUR, which can be moved to your PLOTTAB directory and immediately plotted; see, the PLOTTAB documentation for details.

## PATHC

BALANCE is designed to allow you to see overall features of an entire system. In contrast, PATHC allows you to see the features of individual spatial zones. It will edit pathlength (flux) spectra as a function of energy for individual spatial zones, and allow you to optionally renormalize the results per unit energy and/or unit volume.

Input Parameters: **PATHC.INP**

To use PATHC, edit **PATHC.INP** to define the output you want, be sure that the file you want it to read is named TART.OUT, and type,

PATHC

You will obtain results in a file named PLOTTAB.CUR, which can be moved to your PLOTTAB directory and immediately plotted; see, the PLOTTAB documentation for details.

WARNING - the name of this utility code has been changed to PATHC from the earlier name PATH. The earlier name caused problems on some systems where attempts to run a code named path by typing,

path

resulted in changing the system defined “path”.

## Source Problems

### FLUXEDIT

FLUXEDIT is designed to edit flux and energy deposition for each spatial zone to PLOTTAB input format, which can be used by TARTCHEK to overlay the results on your geometry.

The FLUXEDIT code reads an entire TART output file, TART.OUT, for **one source problem**, and produces output in the PLOTTAB input format, so that you can immediately see your results.

Input Parameters: **NONE**

To use FLUXEDIT, be sure that the file you want it to read is named TART.OUT, and type,

FLUXEDIT

You will obtain results in a file named FLUXEDIT.OUT, which can be moved to your TARTCHEK directory and immediately plotted. See the chapter on **TARTCHEK** in the TART95 documentation to learn how to overlay your results on your geometry.

### PATHS

Similar to PATHC for criticality problems, PATHS allows you to see the features of individual spatial zones for source problems. It will edit pathlength (flux) spectra as a function of energy for individual spatial zones, and allow you to optionally renormalize the results per unit energy and/or unit volume. You can obtain output for neutron and/or photon source problems.

Input Parameters: **PATHS.INP**

To use PATHS, edit **PATHS.INP** to define the output you want, be sure that the file you want it to read is named TART.OUT, and type,

PATHS

You will obtain results in a file named PLOTTAB.CUR, which can be moved to your PLOTTAB directory and immediately plotted; see, the PLOTTAB documentation for details.

### EDIT1112

TART tally types 11 and 12 for neutrons or photons allows users the option of tallying particles when they enter individual spatial zones; the particle is then terminated only if

the zone is empty. For each particle the output includes all spatial, energy, time, direction, and the number of collisions from source. This option is used by many TART users for special applications, such as detector response problems. Starting with the 2000 distribution, tally type 11 or 12 can alternately be used to tally absorptions within zones, rather than particles entering a zone, allowing for improved simulation of detectors.

EDIT1112 was initially designed to read a binary file created by TART and produce tabulated results that users can then use in their applications. However, currently it is more often used by users within their special purpose application codes. EDIT1112 is incorporated into their codes and is used to read the coordinates for one particle at a time from a binary file, and the users then do whatever they want with them.

The EDIT1112 code reads **a single binary** TART output file, for **a single source problem**, and creates a file of tabulated results. The user defines by input the file name of the output tabulated result.

Input Parameters: **EDIT1112.INP**

To use EDIT1112, edit **EDIT1112.INP** to define the output you want, be sure the binary file you want it to read is in the same directory, and type,

EDIT1112

You will obtain tabulated results in the file you specified by input.

**WARNING** - EDIT1112 will only read **a single binary** file, even though TART may create an entire family of binary files. This is done because initial tests of the code illustrated that reading and tabulating the results for an entire family of files can result in enormous output files. To use an entire family of file, see PLOT1112 below.

## PLOT1112

PLOT1112 is merely one example of a special purpose code built on top of EDIT1112. PLOT1112 reads **a family of binary** files created by TART, summarize the results in a file name defined by user input, and also produces results versus time, in a file named **TIME.CUR**, versus energy, in a file named **ENERGY.CUR**, and versus the direction cosine with respect to the Z axis, in a file names **ZCOSINE.CUR**. Each of these files is in the PLOTTAB format, so that you can use PLOTTAB to see your results. Each file contains PLOTTAB "curves" for total results, as well as results for each number of collisions that particles had between source and entering the tally zone, e.g., uncollided, first collided, second collided, etc., results.

Input Parameters: **PLOT1112.INP**

To use PLOT1112, edit **PLOT1112.INP** to define the output you want, be sure the entire family of binary files you want it to read is in the same directory, and type,

PLOT1112

You will obtain a summary of results in the file you specified by input, as well as **TIME.CUR** and **ENERGY.CUR**, that can be used with PLOTTAB to see your results. Note, **TIME.CUR** and **ENERGY.CUR** correspond to the standard PLOTTAB input file **PLOTTAB.CUR**. By moving either to your PLOTTAB directory and renaming it **PLOTTAB.CUR** you can immediately see your results. See, the PLOTTAB documentation for details.